

JOBSHEET 7

QUEUE

Mata Kuliah : Algoritma dan Struktur Data

Dosen : **Mungki Astiningrum, S.T., M.Kom.**



ILHAM DHARMA ATMAJA (244107020220)

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG TAHUN 2025**

1. Praktikum 1

1.1 Program

1) Class Queue14

```
import
java.util.Scanner;
public class Queue14 {
    int data[];    int
    front;    int rear;
    int size;    int max;

    public Queue14(int n)
    {
        max = n;
        data = new int[max];
        size = 0;    front =
        rear = -1;
    }
}
```

2) method IsEmpty dan IsFull

```
public boolean IsEmpty() {

    return size == 0;

}
```

```
public boolean IsFull() {
    return size == max;
```

```
}
```

3) method peek

```

    public void peek() {
    if (!IsEmpty()) {
        System.out.println("Data terdepan: " + data[front]);
    } else {
        System.out.println("Queue masih kosong");
    }
    }
}

```

4) method print

```

public void print() {
    if (IsEmpty()) {
        System.out.println("Queue masih kosong");
    } else {
        int i = front;
        while (i != rear) {
            System.out.print(data[i] + " ");
            i = (i + 1) % max;
        }
        System.out.println(data[i] + " ");
        System.out.println("Jumlah elemen = " + size);
    }
}

```

5) method clear

```

public void clear() {
    if (!IsEmpty()) {
        front = rear = -1;
        size = 0;
        System.out.println("Queue berhasil dikosongkan");
    } else {
        System.out.println("Queue masih kosong");
    }
}

```

6) method Enqueue

```

public void Enqueue(int dt) {
    if (IsFull()) {
        System.out.println("Queue sudah penuh");
    } else {
        if
(IsEmpty()) {
            front = rear = 0;
        } else {
            if (rear == max - 1) {
                rear = 0;
            } else {
                rear++;
            }
        }
        data[rear] =
dt;
        size++;
    }
}

```

7) method Dequeue

```

    public int Dequeue() {
        int dt = 0;          if
        (IsEmpty()) {
            System.out.println("Queue masih kosong");
        } else {            dt =
        data[front];          size--;
        if (IsEmpty()) {      front
        = rear = -1;          } else {
        if (front == max - 1) {
        front = 0;            } else {
        front++;
                                }
                            }
        }
    }      return
    dt;
    }

```

```

    public static void menu()
    {
        System.out.println("Masukkan operasi yang diinginkan:");
        System.out.println("1. Enqueue");
        System.out.println("2. Dequeue");
        System.out.println("3. Print");
        System.out.println("4. Peek");
        System.out.println("5. Clear");
        System.out.println("6. Keluar");
        System.out.println("-----");
    }

```

8) method menu

```

public static void QueueMain14(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.print("Masukkan kapasitas queue: ");
    int n = sc.nextInt();

    Queue14 Q = new Queue14(n);

    int pilih;
    do {
        menu();
        pilih =
        sc.nextInt();
        switch
        (pilih) {
            case 1:
                System.out.print("Masukkan data baru: ");
                int dataMasuk = sc.nextInt();
                Q.Enqueue(dataMasuk);
                break;
            case 2:
                int dataKeluar = Q.Dequeue();
                if (dataKeluar != 0) {
                    System.out.println("Data yang dikeluarkan:
" + dataKeluar);
                    break;
                }
            case 3:
                Q.print();
                break;
            case 4:
                Q.peek();
                break;
            case 5:
                Q.clear();
                break;
            case 6:
                System.out.println("Keluar dari
program.");
                break;
            default:
                System.out.println("Pilihan tidak valid.");
        }
    } while (pilih != 6);
}

```

9) QueueMain

1.2 Verifikasi Hasil Percobaan

```
Masukkan kapasitas queue: 4
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
6. Keluar
-----
1
Masukkan data baru: 15
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
6. Keluar
-----
1
Masukkan data baru: 31
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
6. Keluar
-----
4
Data terdepan: 15
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
6. Keluar
-----
```

1.3 Pertanyaan

1. Front dan rear = -1 menandakan queue kosong sedangkan size = 0 Karena belum ada elemen yang masuk ke queue
2. Jika rear sudah di ujung array, maka Kembali ke awal (0) untuk membuat queue melingkar
3. Jika front sudah di ujung array, maka Kembali ke awal (0) untuk melanjutkan penghapusan elemen di queue melingkar
4. Karena elemen valid queue dimulai dari posisi front, bukan selalu dari indeks 0
5. Untuk melanjutkan ke indeks berikutnya secara melingkar. Jika sudah di ujung array, Kembali ke indeks 0

6.

```
if (IsFull()) {
```

```
    System.out.println("Queue sudah penuh");
```

```
}
```

7.

```
if (IsFull()) {
```

```
    System.out.println("Queue sudah penuh");
```

```
    System.exit(0);
```

```
} if
```

```
(IsEmpty()) {
```

```
    System.out.println("Queue masih kosong");
```

```
    System.exit(0);
```

```
}
```


2. Praktikum 2

2.1 Program

1. Class Mahasiswa

```
public class Mahasiswa14 {  
    String nim;  
    String nama;  
    String prodi;  
    String kelas;  
    public Mahasiswa14(String nim, String nama, String prodi, String kelas)  
{  
        this.nim = nim;        this.nama = nama;        this.prodi =  
prodi;        this.kelas = kelas;  
    }    public void  
tampilkanData() {  
        System.out.println(nim + " - " + nama + " - " + prodi + " - " + kelas);  
    }  
}
```

2. Class AntrianLayanan

```
public class AntrianLayanan14 {
    int max, size, front, rear;
    Mahasiswa14[] data;

    public AntrianLayanan14(int n)
    {
        max = n;
        data =
        new Mahasiswa14[max];
        size =
        0;
        front = rear = -1;
    }

    public boolean
    isEmpty() {
        return
        size == 0;
    }

    public boolean
    isFull() {
        return
        size == max;
    }

    public void
    enqueue(Mahasiswa14 m) {
        if
        (isFull()) {
            System.out.println("Antrian penuh");
            return;
        }

        if
        (isEmpty()) {
            front = rear = 0;
        } else {
            rear++;
        }

        data[rear] = m;
        size++;
    }

    public Mahasiswa14
    dequeue() {
        if (isEmpty())
        {
            System.out.println("Antrian kosong");
```

```

        System.out.println("Antrian kosong");
return null;
    }

    Mahasiswa14 m = data[front];
for (int i = front; i < rear; i++) {
data[i] = data[i + 1];
    }        rear--;
size--;        if (size == 0)
{
        front = rear = -
1;
    }
return m;
    }    public void peek() {
if (!isEmpty()) {
data[front].tampilkanData();
    } else {
        System.out.println("Antrian kosong");
    }
}

public void print() {
if (isEmpty()) {
        System.out.println("Antrian kosong");
return;
    }

    System.out.println("Data antrian:");
for (int i = front; i <= rear; i++) {
data[i].tampilkanData();
        System.out.println("-----");
    }    }    public int
getJumlahAntrian() {        return
size;
    }

```

```

public void lihatAkhir() {
    if (!isEmpty()) {
        System.out.println("Antrian terakhir:");
        data[rear].tampilkanData();
    } else {
        System.out.println("Antrian kosong");
    }
}
}

```

3.TUGAS

1. Class MahasiswaKRS

```

import java.util.Scanner;

public class MainKRS14 {      public static
void main(String[] args) {      Scanner
    sc = new Scanner(System.in);

        AntrianKRS14 antrian = new AntrianKRS14(10);
    int pilihan;

        do
    {

        System.out.println("\n=== MENU ANTRIAN KRS ===");
        System.out.println("1. Tambah Mahasiswa ke Antrian");
        System.out.println("2. Proses Antrian Mahasiswa");
        System.out.println("3. Lihat Antrian Terdepan");
        System.out.println("4. Lihat Antrian Terakhir");
        System.out.println("5. Lihat Semua Antrian");
        System.out.println("6. Lihat Jumlah Antrian");
        System.out.println("0. Keluar");
        System.out.print("Pilih menu: ");
        pilihan = sc.nextInt();          sc.nextLine();
    }
}

```

```

switch (pilihan) {
case 1:

        System.out.print("NIM: ");

        String nim = sc.nextLine();

        System.out.print("Nama: ");

        String nama = sc.nextLine();

System.out.print("Semester: ");                                int
semester = sc.nextInt();
sc.nextLine();

        MahasiswaKRS14 m1 = new MahasiswaKRS14(nim, nama, semester);
antrian.enqueue(m1);                                break;

        case

2:

        MahasiswaKRS14 m2 = antrian.dequeue();

if (m2 != null) {

        System.out.println("Mahasiswa diproses untuk KRS:");

m2.tampilkanData();

        }

break;

        case

3:

        antrian.lihatTerdepan();

break;

        case

4:

        antrian.lihatTerbelakang();

break;

        case

5:

        antrian.tampilkanAntrian();

break;

        case

6:

        System.out.println("Jumlah mahasiswa dalam antrian: " +
antrian.jumlahAntrian());                                break;

case 0:

```

```
case 0:

        System.out.println("Terima kasih!");

break;

default:

        System.out.println("Pilihan tidak valid.");

break;

        }

    } while (pilihan != 0);

}

}
```

2. CLASS AntrianKRS

```

public class AntrianKRS14 {
    int max, size, front, rear;

    MahasiswaKRS14[] data;

    public AntrianKRS14(int kapasitas)
    {
        max = kapasitas;        data
    = new MahasiswaKRS14[max];    size
    = 0;        front = rear = -1;

        }        public boolean
isEmpty() {        return
size == 0;

        }        public boolean
isFull() {        return
size == max;

        }        public void
enqueue(MahasiswaKRS14 m) {        if
(isFull()) {

            System.out.println("Antrian KRS penuh!");

```

```

        return;

```

```

        return;
    }

    if (isEmpty()) {
        front = rear = 0;
    } else {
        rear++;
    }

    data[rear] = m;
    size++;

    System.out.println("Mahasiswa berhasil ditambahkan ke antrian KRS.");
}

public MahasiswaKRS14 dequeue() {
    if (isEmpty()) {
        System.out.println("Antrian KRS kosong!");
        return null;
    }

    MahasiswaKRS14 m = data[front];
    for (int i = front; i < rear; i++) {
        data[i] = data[i + 1];
    }
    rear--;
    size--;
    if (size == 0) {
        front = rear = -1;
    }

    return m;
}

public void tampilkanAntrian() {
    if (isEmpty()) {
        System.out.println("Antrian kosong.");
        return;
    }

    System.out.println("Daftar Antrian KRS:");
    for (int i = front; i <= rear; i++) {

```



```

        System.out.println("Antrian ke-" + (i - front + 1));
data[i].tampilkanData();

        System.out.println("-----");
    }

    public int
jumlahAntrian() {
return size;

    } public void
lihatTerdepan() {
if (!isEmpty()) {
        System.out.println("Mahasiswa pada antrian terdepan:");
data[front].tampilkanData();
    } else {
        System.out.println("Antrian kosong.");
    }
    } public void
lihatTerbelakang() {
if (!isEmpty()) {
        System.out.println("Mahasiswa pada antrian terakhir:");
data[rear].tampilkanData();
    } else {
        System.out.println("Antrian kosong.");
    }
    }
}

```

3. CLASS MainKRS

```
import java.util.Scanner;

public class MainKRS14 {    public static
void main(String[] args) {        Scanner
sc = new Scanner(System.in);

        AntrianKRS14 antrian = new AntrianKRS14(10);
int pilihan;

        do

{

        System.out.println("\n=== MENU ANTRIAN KRS ===");
        System.out.println("1. Tambah Mahasiswa ke Antrian");
        System.out.println("2. Proses Antrian Mahasiswa");
        System.out.println("3. Lihat Antrian Terdepan");
        System.out.println("4. Lihat Antrian Terakhir");
        System.out.println("5. Lihat Semua Antrian");
System.out.println("6. Lihat Jumlah Antrian");

        System.out.println("0. Keluar");
System.out.print("Pilih menu: ");
pilihan = sc.nextInt();
sc.nextLine();    switch (pilihan) {
case 1:

                System.out.print("NIM: ");
                String nim = sc.nextLine();
                System.out.print("Nama: ");
                String nama = sc.nextLine();
System.out.print("Semester: ");                                int
semester = sc.nextInt();
sc.nextLine();

                MahasiswaKRS14 m1 = new MahasiswaKRS14(nim, nama,
semester);                                antrian.enqueue(m1);                                break;
```

```

        case 2:
            MahasiswaKRS14 m2 = antrian.dequeue();
            if (m2 != null) {
                System.out.println("Mahasiswa diproses untuk
KRS:");
                m2.tampilkanData();
            }
            break;

            case
3:
            antrian.lihatTerdepan();
            break;

            case
4:
            antrian.lihatTerbelakang();
            break;

            case
5:
            antrian.tampilkanAntrian();
            break;

            case
6:
            System.out.println("Jumlah mahasiswa dalam antrian:
" + antrian.jumlahAntrian());
            break;

            case
0:
            System.out.println("Terima kasih!");
            break;

        default:
            System.out.println("Pilihan tidak valid.");
            break;
    }
} while (pilihan != 0);
}
}

```

3.1 Verifikasi Hasil

```
=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses Antrian Mahasiswa
3. Lihat Antrian Terdepan
4. Lihat Antrian Terakhir
5. Lihat Semua Antrian
6. Lihat Jumlah Antrian
0. Keluar
Pilih menu: 1
NIM: 12345
Nama: koko
Semester: 2
Mahasiswa berhasil ditambahkan ke antrian KRS.

=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses Antrian Mahasiswa
3. Lihat Antrian Terdepan
4. Lihat Antrian Terakhir
5. Lihat Semua Antrian
6. Lihat Jumlah Antrian
0. Keluar
Pilih menu: 1
NIM: 23456
Nama: keni
Semester: 2
Mahasiswa berhasil ditambahkan ke antrian KRS.
```

```
=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses Antrian Mahasiswa
3. Lihat Antrian Terdepan
4. Lihat Antrian Terakhir
5. Lihat Semua Antrian
6. Lihat Jumlah Antrian
0. Keluar
Pilih menu: 5
Daftar Antrian KRS:
Antrian ke-1
NIM      : 12345
Nama     : koko
Semester : 2
-----
Antrian ke-2
NIM      : 23456
Nama     : keni
Semester : 2
-----
```

3.2 LINK GIT HUB

<https://github.com/ianmen10/SEMESTER-genap2.git>